

EEE 304 Lab: Control (Extra Credit)

1. Bill of Materials

In this laboratory experiment, you will build and test a simple feedback control experiment. The idea is to maintain a constant light in a space as measured by a sensor. Think of this as a watered down version of illumination control, wherein the amount of light inside a room is kept constant throughout the day. In our case, we will only use one light source and one sensor.

In order to carry out this experiment you will be required to obtain the following

Table 1: List of parts to be purchased

Item	Model	Quantity
Arduino	Due	1
Micro USB cable	Any android phone charger	1
Photo Transistor	3034a	2
Bright White LED	12900 mcd WHITE LED	1
Resistor	100 K Ohm	1
Set of wires	Male to Male Jumper wires	10
Breadboard	Any breadboard	1

You will require at least the quantities mentioned in the table to successfully carry out the experiment. The hyperlinks attached to the **item names** are for example only and you can purchase these items from anywhere you like.

You will also need MS Visual Studio to work as a compiler, instructions on how to install is posted on Blackboard.

If you haven't completed the first extra credit lab assignment, you need to go through the following sections in the first extra credit lab to make sure software is setup correctly:

Section 2 "Introduction to Arduino DUE"

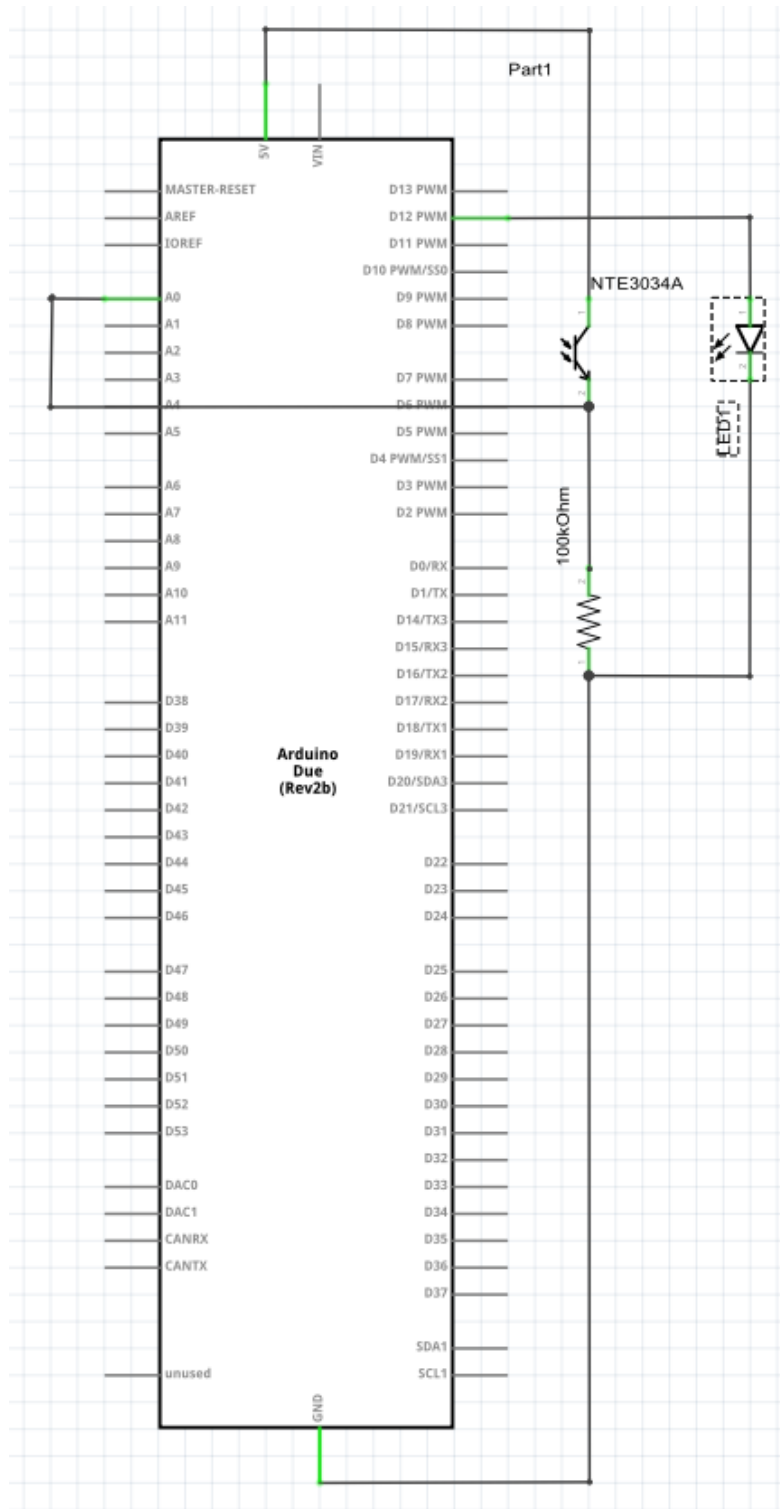
Section 3 "Software Preparation"

Section 4.2 Building the Simulink Test Model

2. Lab Procedure

2.1 Building the Circuit

Using the parts purchased, connect the circuit as shown in Figure 1. If you are unsure of the pin labels or for any details, you may look up datasheet for the parts online.



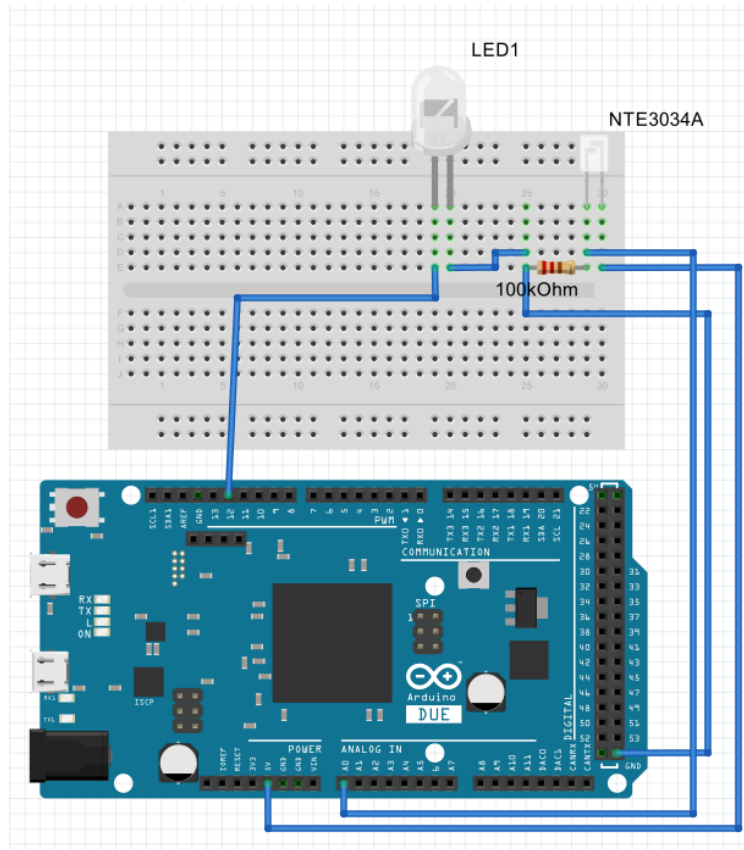


Figure 1: Building the circuit.

The phototransistor has two pins “emitter” (E) and “collector” (C). C goes to 5v on the Arduino and E goes to the resistor. The LED recommended emits plenty of light and is rated for 5v. So, plugging it in to the 3.3v PWM pin 12 of the Arduino will not damage the LED. A resistor may be placed between the LED and GND pin of the Arduino but that will limit current and therefore light output. You may add a resistor if the LED you purchased burns out.

Another thing to keep in mind, when you place the LED and the Photoresistor on the breadboard, make sure the LED is facing the Phototransistor and that they are not too far apart. Preferably less than 1 cm distance between the two. Otherwise the light intensities may not be read properly.

2.2 Identifying the Plant

For the purpose of control, we need a model for our system. Since this is a fairly simple experiment the plant will be a gain; no dynamics is present, therefore no poles or zeros. Also, due to the nature of the problem the plant initially is highly nonlinear. This will make control difficult so we will perform an experiment to linearize the plant and compute the gain of the system. In our plant, the output is PWM voltage value which controls light intensity, and the input is light received measured as a change in current which induces a voltage on the 100k Ohm resistor.

Open “**Calibrate_sensor.slx**” model file. The **manual switch**, allows us to run between linearized and non-linearized model of the plant. Double-clicking the switch allows to change between the two input ports. Keep the switch in the top position and run the model. The **slider** allows you to change the plant input, alternatively you could double-click on the **constant block** and change its value there as well. Observe, that as the values are changed between 10 – 255 in the constant block, the display block labeled “light intensity” will go between 0 – 1023. The

Analog input of the Due is utilizing 10 bits of data that represents a measured voltage between 0-3.3V.

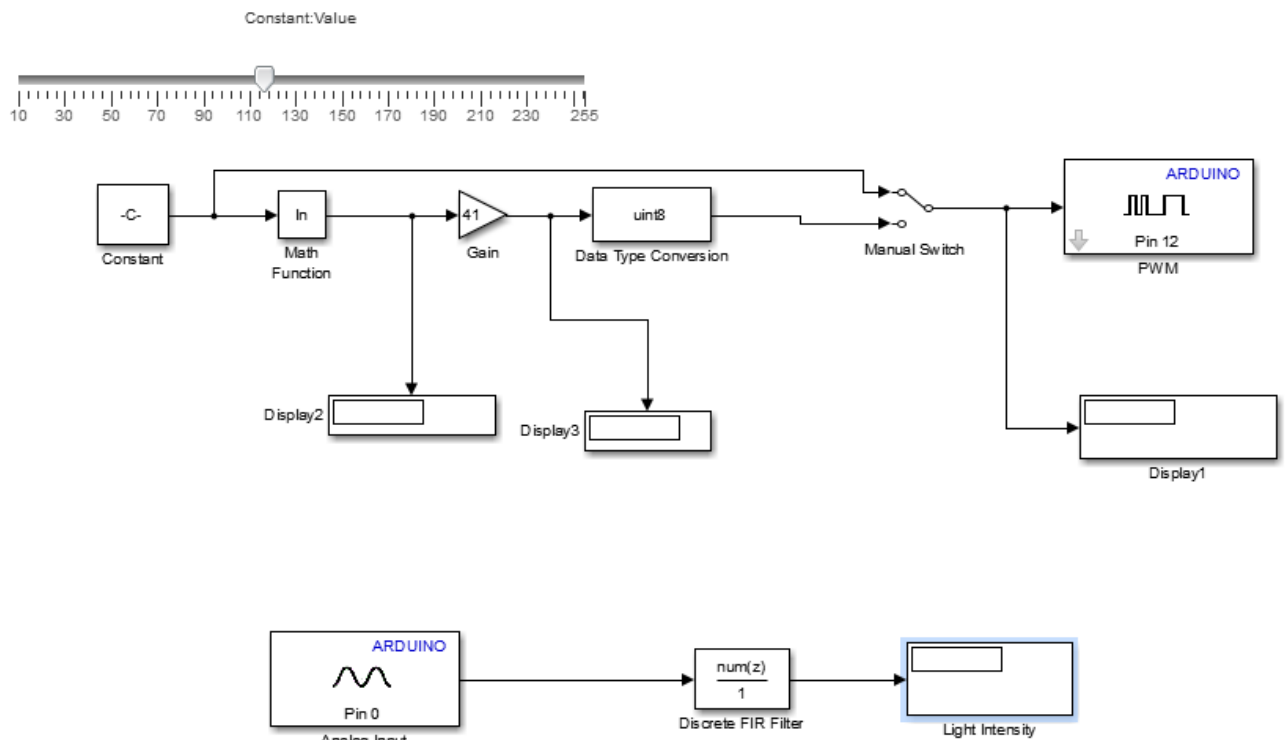


Figure 2: Identifying the plant.

TASK 1: As part of your report, create a table as shown below. Change the value of the plant input and record the output value in the table below.

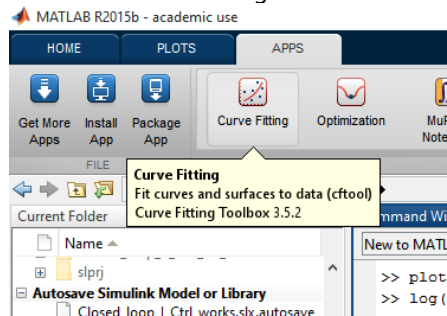
Input(constant block)	Output (Light Intensity)
10	
30	
50	
70	
90	
110	
130	
150	
170	
190	
210	
220	
224	
226	
228	
230	

TASK 2: As part of your report, include a plot of this input-output data. You can use any software you like to do so; however, MATLAB is strongly recommended. To plot in MATLAB, create a variable called "inp" as shown

```
>> inp = [10;30;50;70;90;110;130;150;170;190;210;220;224;226;228;230]
```

Likewise, create another output called "outp". With these variables in the workspace the data can be plotted with the "plot" command.

You must now fit this data using the curve fitting toolbox in order to linearize the plant. Once again, any math software can be used but this manual will utilize MATLAB. As shown in the image below, go to "Apps" and select "curve fitting".



The curve fitting app will start up as shown in Fig. 3.

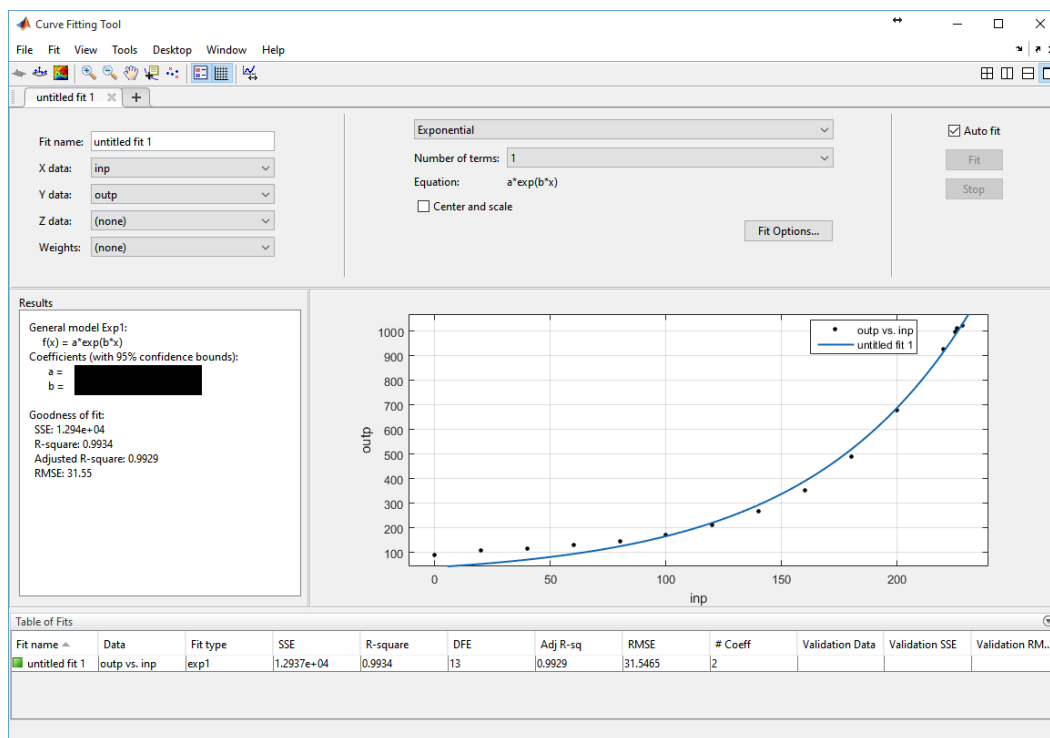


Figure 3: Curve fitting.

Choose, the appropriate "x" and "y" data (inp and outp, respectively) and then make sure you choose exponential fitting. The tool will fit the data as shown and report the values of "a" and "b" that fits the data into an exponential equation of form: $outp = a e^{inp \times b}$.

TASK 3: As part of your report, provide a screenshot of the curve-fitting and the parameters "a" and "b".

Now, we are ready to linearize the plant. Since the input-output behavior of our plant is exponential. Taking a logarithm on the output linearize it, i.e., $\ln(outp) = \ln(a) + inp \times b$. However, we need to scale the output appropriately. This is achieved by the "gain" block. Modify the gain in the Simulink gain block to the value of "a" computed from curve fitting.

TASK 4: As part of your report, you must once more fill out a second copy of the table from Task 1, but this time change the "manual switch" to the second port so that the input is now

linearized. Provide a plot of the input vs output curve (as in Task 2) and show that it is linear. Provide the gradient of the curve. How does it compare to the value of “a” computed in Task 3?

2.3 Closed-loop Control

Now that the plant is linearized and identified, we can start designing a controller. In our case, the plant, $P(s) = a$. So a simple integral controller will be sufficient to control the plant and keep the light intensity as received by the phototransistor constant. Then, the controller is $C(s) = \frac{k}{s}$. Forming the open-loop transfer function we get $L(s) = \frac{ka}{s}$. As can be seen by looking at the bode plot of such a transfer function, the plant has 90° phase margin for all gains. Let us start with a gain of 0.1 and simulate the closed-loop response. Open “**Closed_loop_I_Ctrl.slx**”, Figure 4.

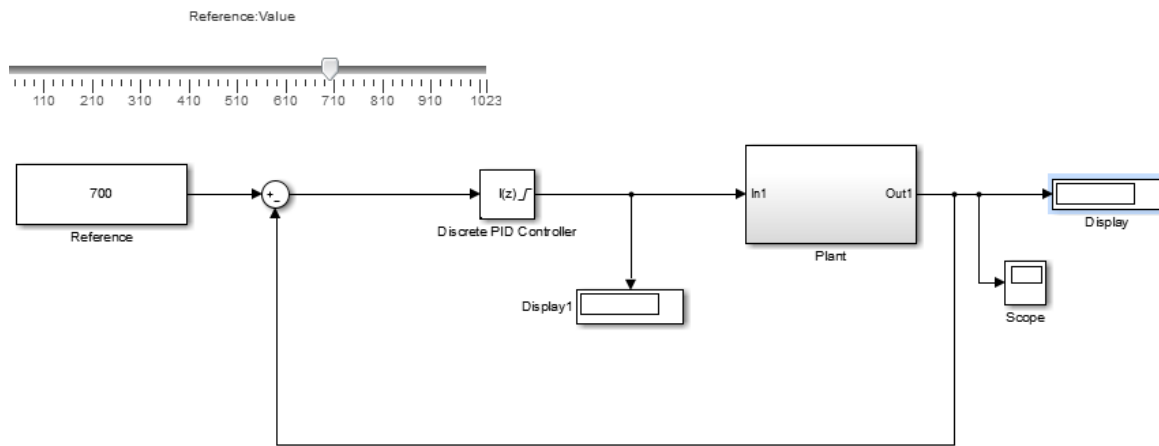


Figure 4: Integral control.

The first step is to calibrate the “Plant” block to the gain of “a” computed in Task 3. Double click the “Plant” block and modify the “gain”. You can go back up to the main model diagram by clicking its name in the “explorer bar”.

The discrete time Integral controller is already configured with a gain of 1 as shown in Fig. 5.

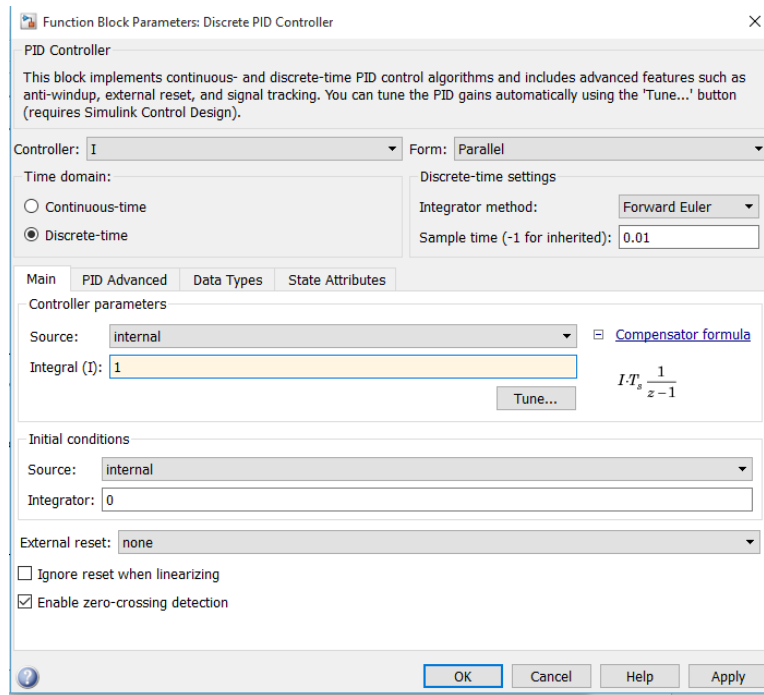
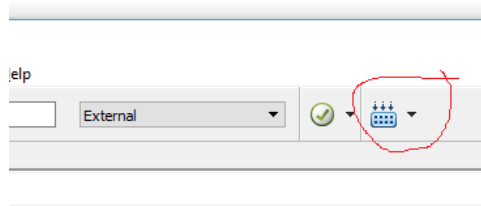


Figure 5: Integral (I) gain.

Now, run the simulation.

TASK 5: As part of your report, provide a screenshot of the scope. Does the output stabilize at the same value of the reference? Is it a smooth curve? If not, explain why and how it could be smoothed.

As mentioned earlier, the quirks of using external mode is that the simulation will not run for long when data is being logged at the same time. Now you will build and download the model onto the Arduino. This time, no data will be logged, the code will run as long as the Arduino is powered. To do this you must click on the "Deploy to hardware" button in the model window as shown below.



This will compile and download the code onto the Arduino without External mode operation. Try applying different light conditions to your model, e.g. making the room dark or applying a torch to the sensor and observe how the LED reacts.

TASK 6: As part of your report, change the gain of the integral controller to 0.1 in Fig. 5. Run the simulation in "External Mode". Provide a screenshot of the scope. Next, run the model in "Deploy to hardware" mode and observe how the LED reacts to changes in light. Compare and comment on how the light reacts when the Integral gain is 1 vs 0.1.

2.4 Closed-loop Instability

Given, the plant model and controller designed in Section 2.3, the root locus suggests that the plant will be stable no matter how high the gain of the system is made. The root locus of an open loop transfer function $L(s)$ can be computed in MATLAB using the `rlocus` command as shown.

```
>> rlocus(L)
```

This argument is quite fallacious as we will see next. For any system, the gain of a controller cannot be arbitrarily increased. In assuming $P(s) = a$, we erroneously assumed that the plant is but a gain; however as we shall see, ALL systems have high frequency dynamics. In the case of this simple electronic circuit, it is the impedance created by parasitic capacitance in the wires. Thus a more reliable model of the system would be $P(s) = a \left(\frac{p}{s+p} \right)^2$, where p is a large number. Although computing the value of p is beyond the scope of this manual, we will attempt to see that arbitrarily increasing gain of a closed-loop system will cause the system to become unstable.

TASK 7: As part of your report, increase the gain of the integral controller in Fig. 5 from 1 in increment of 1 and run the model in "External Mode". Observe the response on the scope. At a certain value of the gain, the output will become completely oscillatory and never stabilize to the reference value. That is the value of the gain that makes the system unstable. Fine tune the gain to 1 decimal place by reiterating the experiment. Provide the value of the integral gain that makes the system unstable as well as a screenshot of the scope.

In conclusion, you have seen how a simple plant can be controlled via feedback. Also, you have observed that a controller is capable of rejecting disturbances (changes in light) from the surrounding. At last, you have also learned that arbitrary design of controllers is a bad idea!

TASK 8: Before you rip everything apart, take a photo of your hardware set up (Arduino DUE, the LED, phototransistor, and wiring) and attach it to the lab report to serve as a proof of you completed the experiment.

EXTRA CREDIT LAB 2: LAB REPORT GRADE SHEET

Name _____

Instructor Assessment

Grading Criteria	Max Points	Points Lost
Template		
Neatness, Clarity, and Concision	2	
Lab Description	8	
Description of Assigned Tasks, Work Performed & Outcomes Met		
Task 1	5	
Task 2	5	
Task 3	5	
Task 4	5	
Task 5	5	
Task 6	5	
Task 7	5	
Task 8	5	
Lab Score (out of 50)	Points Lost	
	Late Lab	
	Lab Score	